Gráficos com ggplot2

```
# bibliotecas utilizadas
if (!"Hmisc" %in% installed.packages()) install.packages("Hmisc")
if (!"ggcorrplot" %in% installed.packages()) install.packages("ggcorrplot")

library(tidyverse)
library(lubridate)
library(magrittr)
library(Hmisc)
```

Por que visualizar os dados?

- Quarteto de Anscombe e a importância da visualização
- Datasaurus Dozen

O que é análise exploratória com gráficos?

Wikipedia

- Forma de responder visualmente a questões levantadas sobre propriedades e relacionamentos entre variáveis de um *dataset*.
- Complemento visual para as estatísticas descritivas.
- Fase de descoberta de problemas, padrões e relacionamentos.
- Fundamental para modelagem e para a comunicação.

ggplot2 (1)

Biblioteca gráfica baseada na gramática dos gráficos composta em camadas.

- Conecta variáveis de Data Frames com elementos gráficos através de abstrações que tornam a visualização independente dos dados tabulares.
- Resolve de forma simples muitas das complexidades da criação de gráficos, como posicionamento de legendas, escalas de cores e formatação de textos em gráficos.
- Expõe um modelo de composição em camadas que facilita o enriquecimento visual através da adição de camadas.

ggplot2 (2)

Cumpre um papel dual:

- Fornece resultados gráficos de alta qualidade, utilizado em publicações e sites de notícias
  - FiveThirtyEight, New York Times são exemplos
- Possibilita a criação de visualizações rápidas para análise exploratória com poucas linhas

O modelo de adição de camadas permite enriquecer um gráfico exploratório simples para ter qualidade de publicação.

```
## # A tibble: 2,550 x 11
##    title  views comments duration               event film_date  languages
##    <chr>  <int>    <int> <S4: Duration>         <fct> <date>         <int>
##  1 Do s~ 4.72e7     4553 1164s (~19.4 minutes)  TED2~ 2006-02-25        60
##  2 Aver~ 3.20e6      265 977s (~16.28 minutes)  TED2~ 2006-02-25        43
##  3 Simp~ 1.64e6      124 1286s (~21.43 minutes) TED2~ 2006-02-24        26
##  4 Gree~ 1.70e6      200 1116s (~18.6 minutes)  TED2~ 2006-02-26        35
##  5 The ~ 1.20e7      593 1190s (~19.83 minutes) TED2~ 2006-02-22        48
##  6 Why ~ 2.07e7      672 1305s (~21.75 minutes) TED2~ 2006-02-02        36
##  7 Lett~ 3.77e6      919 992s (~16.53 minutes)  TED2~ 2006-02-24        31
##  8 Behi~ 9.68e5       46 1198s (~19.97 minutes) TED2~ 2006-02-23        19
##  9 Let'~ 2.57e6      852 1485s (~24.75 minutes) TED2~ 2006-02-02        32
## 10 A li~ 3.10e6      900 1262s (~21.03 minutes) TED2~ 2006-02-25        31
## # ... with 2,540 more rows, and 4 more variables: main_speaker <chr>,
## #   num_speaker <int>, published_date <dttm>, speaker_occupation <fct>
```

Componentes mínimos necessários

O `ggplot`, assim como o `dplyr`, utiliza uma DSL (Domain Specific Language, Linguagem Específica de Domínio) que descreve seus componente.

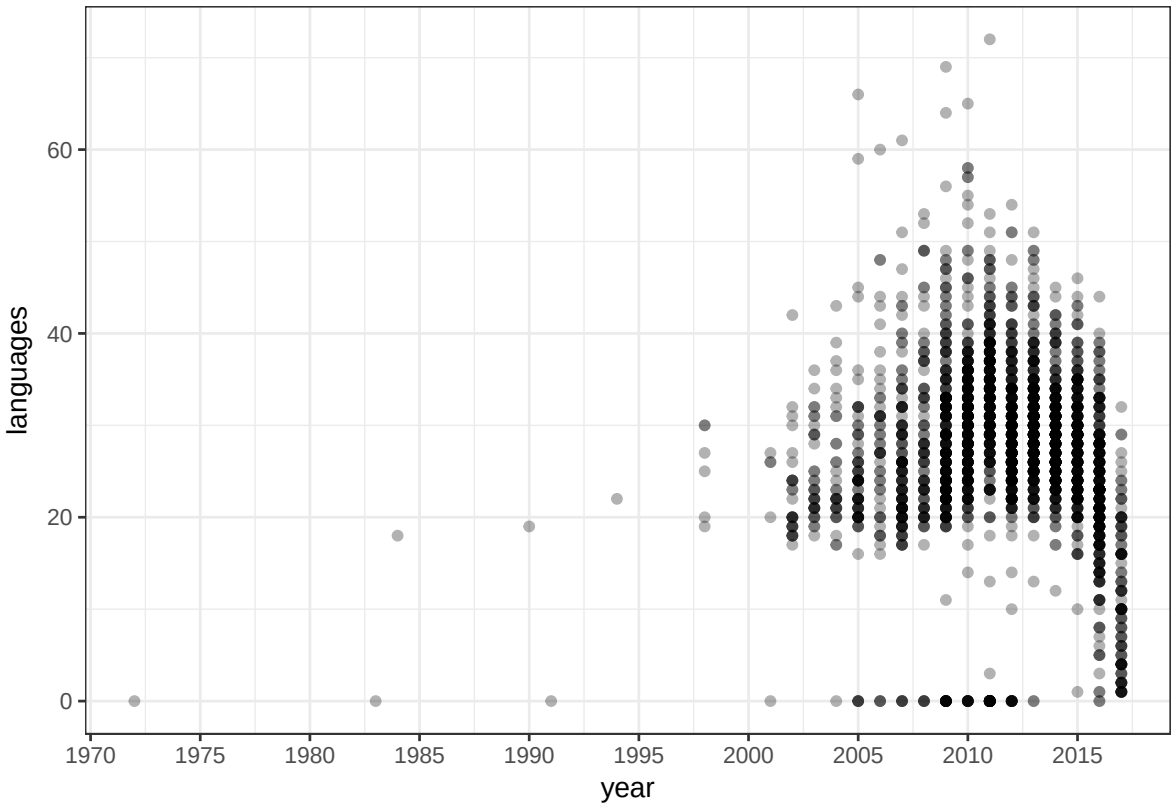
Os principais conceitos do `ggplot` são:

- **canvas**, espaço visual onde formas geométricas serão exibidas
- **estéticas**, que são propriedades visuais dos elementos gráficos
- **mapeamento de estéticas**, que conecta as propriedades visuais com variáveis dos data frames
- **geometrias**, formas geométricas exibidas no canvas
- **escalas**, que são controles visuais das variáveis mapeadas para estéticas
- **tema**, que define atributos visuais do canvas

Tomando como exemplo um gráfico de pontos, vamos visualizar a quantidade de linguagens por vídeo ao longo dos anos. Para este gráfico escolhi utilizar a data de filmagem.

- Iniciamos com a criação da variável ano a partir da variável film_date. O data frame resultante desta transformação é conectado às propriedades visuais por meio do mapeamento de estéticas, que conecta a variável year com o eixo **x** e a variável languages com o eixo **y**. Estas variáveis serão utilizadas como *default* em todas as formas geométricas deste gráfico.
- A seguir, utilizamos a forma geométrica do ponto. Neste exemplo modificamos a propriedade *alpha* para adicionar transparência ao preencimento do ponto. Esta transparência facilita a identificação visual da concentração de pontos em um mesmo par (x, y)
- A escala x é modificada para que os rótulos exibam os anos de 5 em 5.
- Escolho o tema theme_bw, que utiliza padrões de preto e branco.

```
ted_talks %>%
  mutate( year = year( film_date )) %>%
ggplot( aes( x = year, y = languages )) +
  geom_point( alpha = .3 ) +
  scale_x_continuous( breaks = seq( from = 1970, to = 2020, by = 5 )) +
  theme_bw()
```



Análise do gráfico

O que identificamos em relação aos mínimos e máximos? Onde temos maior ocorrência de apresentações, nos eixos x e y? Que padrões a transparência destaca?

Rótulos

No ggplot2 os rótulos podem ser inseridos de diferentes formas. A forma mais consistente é através da função **labs**, que possibilita informar o rótulo de cada estética. No exemplo abaixo atualizei os rótulos dos eixos **x** e **y**, e aproveitei para inserir títulos no gráfico.

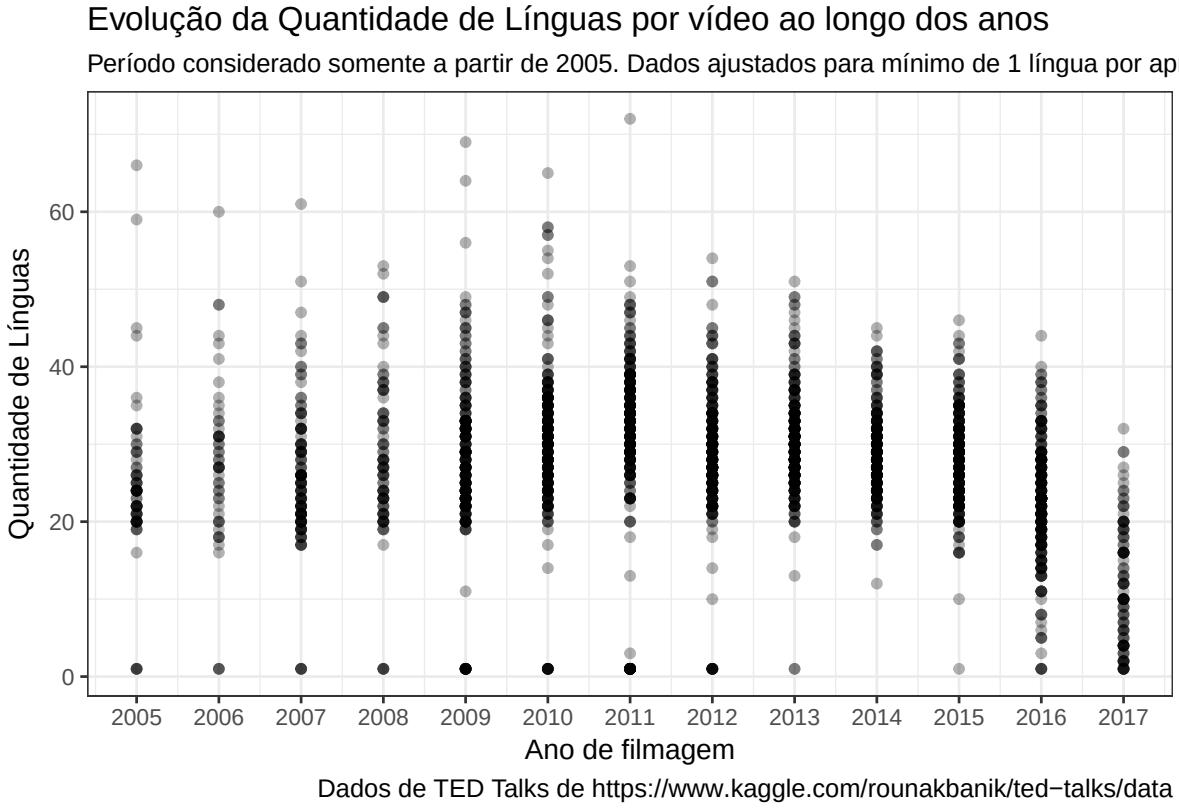
Seguindo com o exemplo de Quantidade de Línguas por ano, vamos reduzir o período para considerar somente apresentações de 2005 em diante. Vídeos que estavam sem quantidade de línguas foram modificados para ter 1 língua.

Pelo padrão abaixo, a decisão de inserir artificialmente 1 língua pareceu acertada?

```
ted_talks_recentes <- ted_talks %>%
  filter(film_date >= ymd(20050101)) %>%
  mutate(languages = if_else(languages == 0, 1L, languages))

ted_talks_recentes %>%
  mutate( year = year( film_date )) %>%
ggplot( aes( x = year, y = languages )) +
  geom_point( alpha = .3 ) +
  scale_x_continuous( breaks = 2005:2017) +
  labs( x = "Ano de filmagem"
      , y = "Quantidade de Línguas"
```

```
      , title = "Evolução da Quantidade de Línguas por vídeo ao longo dos anos"
      , subtitle = "Período considerado somente a partir de 2005. Dados ajustados para mínimo de 1 língu
      , caption = "Dados de TED Talks de https://www.kaggle.com/rounakbanik/ted-talks/data") +
  theme_bw()
```



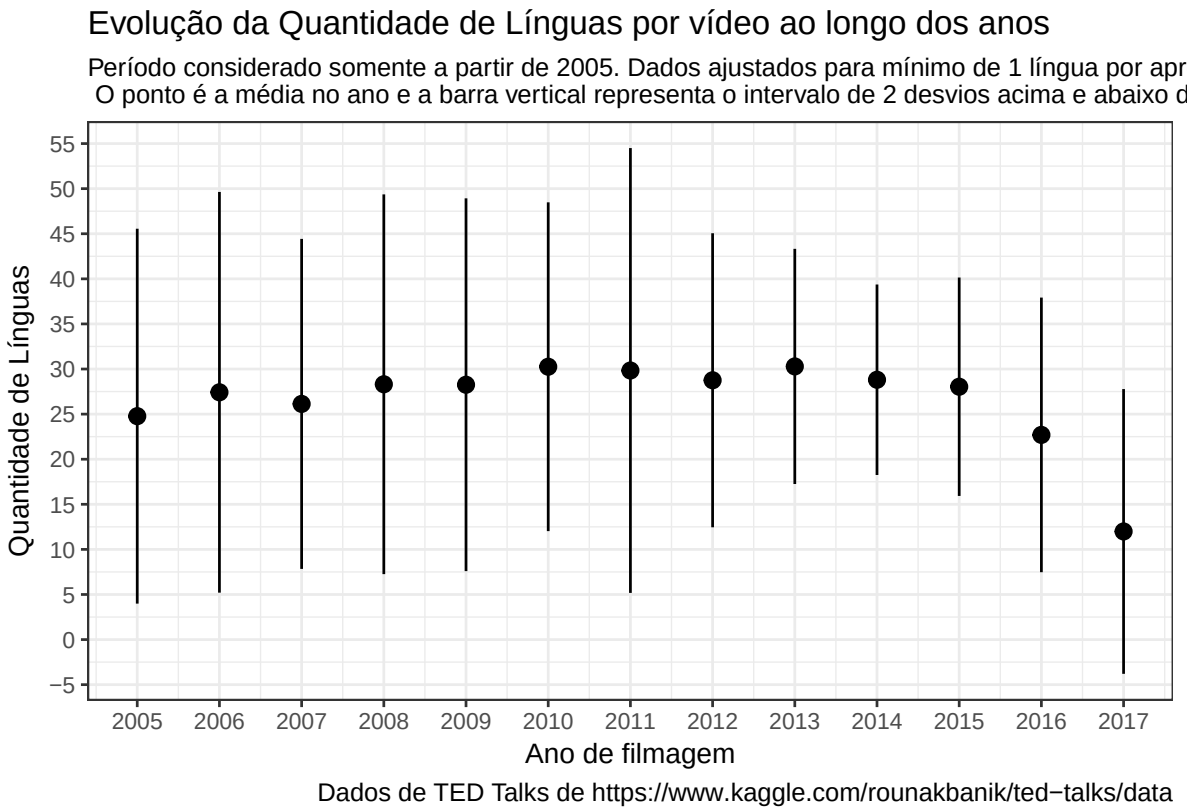
Estatísticas

Estatísticas são combinações de formas geométricas que apresentam visualmente o resultado de estatísticas aplicadas sobre grupos. No exemplo abaixo substituimos a forma de pontos pela forma de resumo `stat_summary`. Esta forma requer uma função que será aplicada sobre a estética **y** para dela derivar um novo **y** central, e mais as estéticas **ymin** e **ymax**. A função escolhida neste caso foi a função `mean_sdl` que retorna o **y** central como a média, o **ymin** como 2 desvios padrão abaixo da média e **ymax** como 2 desvios padrão acima da média.

- Incluí uma escala para o eixo y
- Reparem que é possível acrescentar quebras nos textos. Neste caso inseri uma quebra de linha no subtítulo

```
ted_talks_recentes %>%
  mutate( year = year( film_date )) %>%
ggplot( aes( x = year, y = languages )) +
  stat_summary(fun.data = mean_sdl) +
  scale_x_continuous( breaks = 2005:2017 ) +
  scale_y_continuous( breaks = seq(from = -10, to = 60, by = 5 )) +
  labs( x = "Ano de filmagem"
      , y = "Quantidade de Línguas"
      , title = "Evolução da Quantidade de Línguas por vídeo ao longo dos anos"
      , subtitle = "Período considerado somente a partir de 2005. Dados ajustados para mínimo de 1 língu
```

```
    , caption = "Dados de TED Talks de https://www.kaggle.com/rounakbanik/ted-talks/data") +
  theme_bw()
```



ATIVIDADE

Repetir os gráficos de pontos e de sumário utilizando o ano de publicação no eixo x e a duração no eixo y. Cuidado com a escala do eixo y!

FIM ATIVIDADE

Gráficos de barras

Diferentes formas geométricas do `ggplot` resultam em barras:

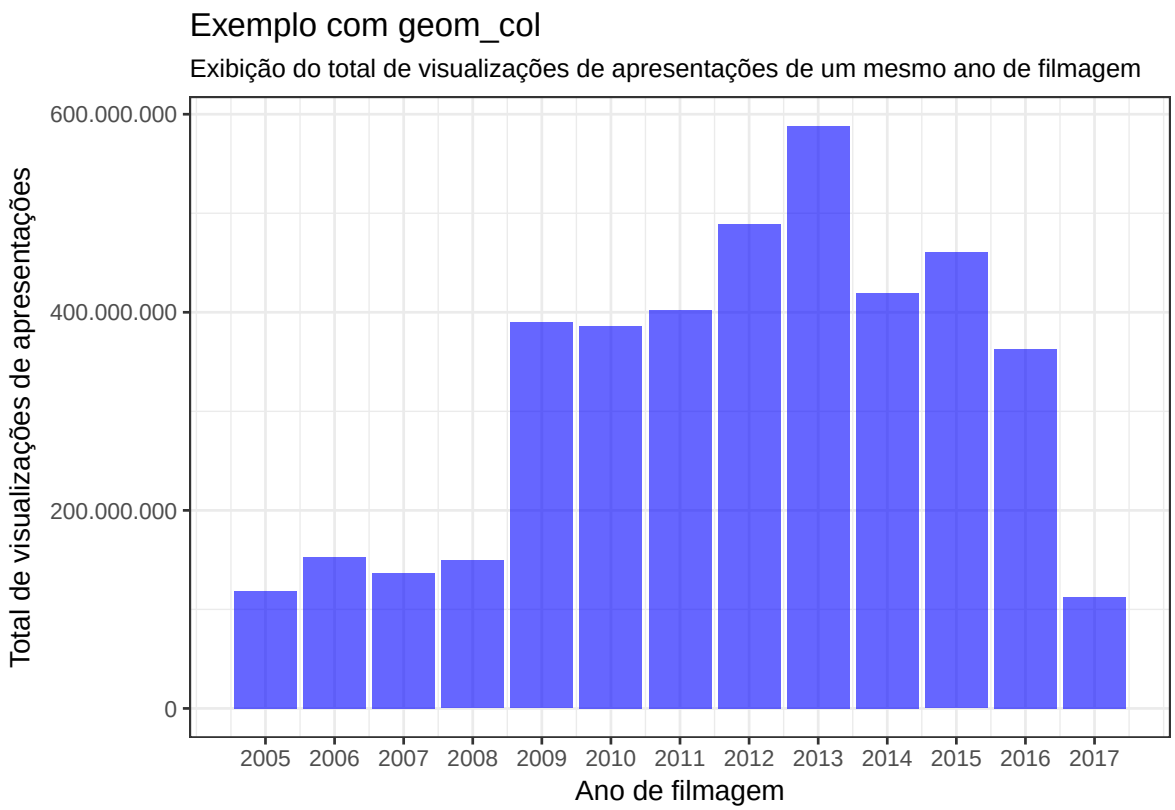
- `geom_col`, quando uma variável do data frame representa o tamanho da barra. Requer as estéticas **x** e **y**.
- `geom_bar`, quando o tamanho da barra for a contagem de observações. Requer a estética **x**.

Exemplo com geom_col

Neste exemplo vemos que é possível determinar a cor de preenchimento da barra através da estética **fill**. Vemos também que é possível formatar um eixo em milhares utilizando a função `format_format` do pacote `scales`.

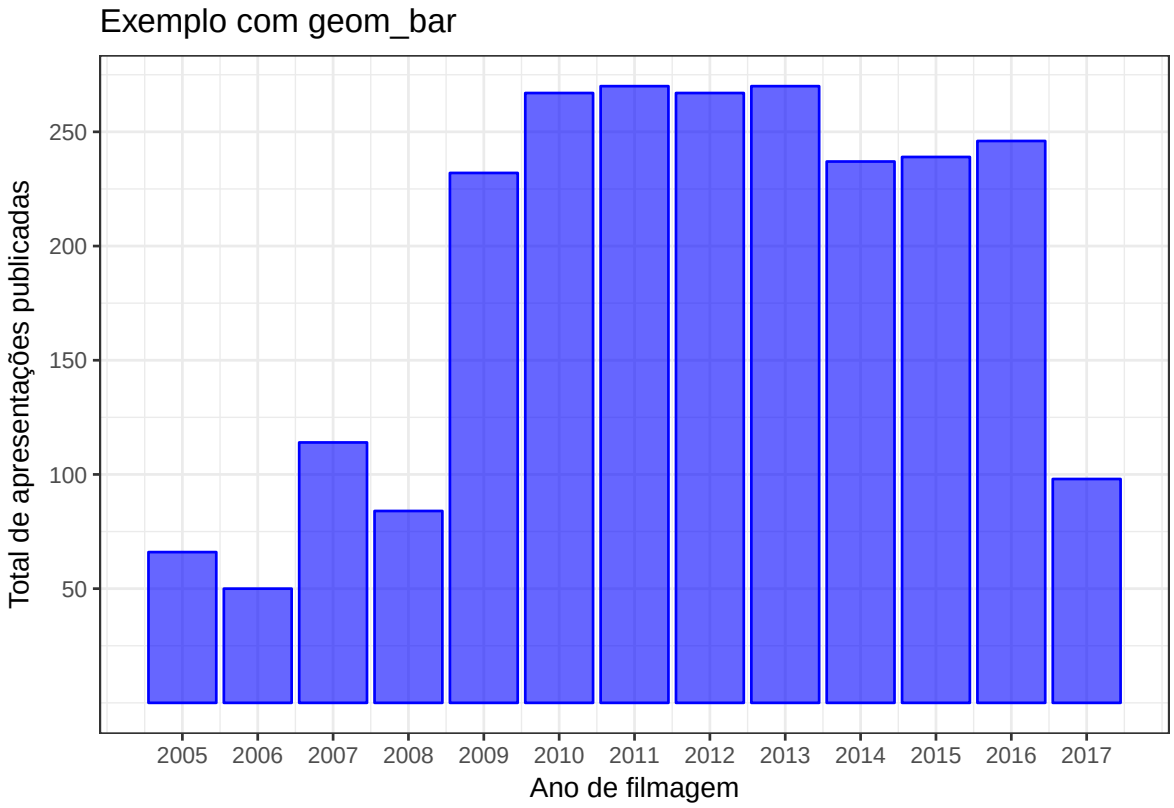
```
ted_talks_recentes %>%
  mutate( year = year( film_date )) %>%
  group_by(year) %>%
  summarise(sum_views = sum(views)) %>%
  ungroup() %>%
```

```
ggplot( aes( x = year, y = sum_views )) +
  geom_col(fill="blue", alpha=0.6) +
  scale_x_continuous(breaks = 2005:2017) +
  scale_y_continuous(labels = scales::format_format(big.mark = ".", decimal.mark=",", scientific = FALSE
  labs( x = "Ano de filmagem"
      , y = "Total de visualizações de apresentações"
      , title = "Exemplo com geom_col"
      , subtitle = "Exibição do total de visualizações de apresentações de um mesmo ano de filmagem") +
  theme_bw()
```



Exemplo com geom_bar

```
ggplot(ted_talks_recentes,  aes( x = year( film_date ))) +
  geom_bar( fill="blue", color = "blue", alpha=0.6 ) +
  scale_x_continuous( breaks = 2005:2017 ) +
  scale_y_continuous( breaks = seq( from = 50, to = 300, by = 50 )) +
  labs( x = "Ano de filmagem"
      , y = "Total de apresentações publicadas"
      , title = "Exemplo com geom_bar" ) +
  theme_bw()
```

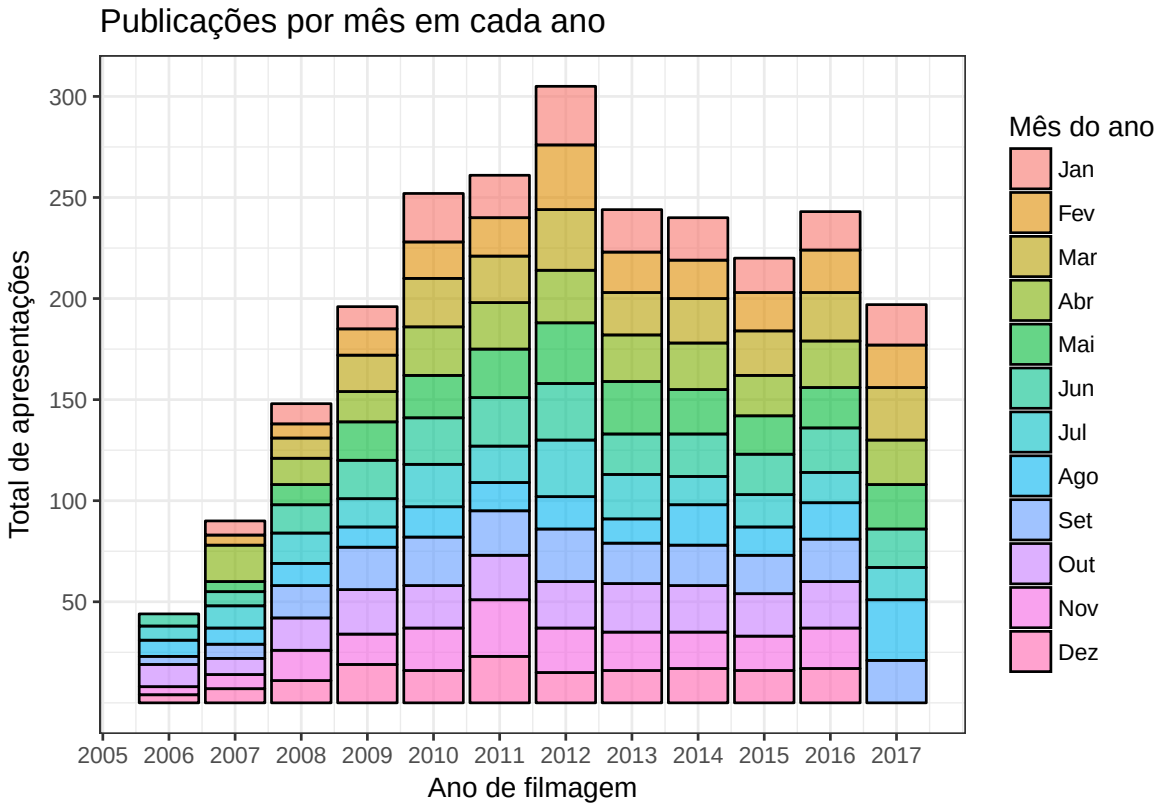


Cores e grupos

No ggplot podemos determinar a cor de uma forma geométrica a partir do mapeamento de uma estética. As cores podem ser em escala contínua quando a variável for numérica e em paletas de cores quando a variável for categórica.

No exemplo abaixo, distiguimos a quantidade de apresentações publicadas por mês utilizando diferentes cores.

```
ted_talks_recentes %>%
  mutate( ano = year( published_date ), mes = month( published_date, label = TRUE )) %>%
ggplot(aes( x = ano, fill = mes )) +
  geom_bar( alpha=0.6, color="black" ) +
  scale_x_continuous( breaks = 2005:2017 ) +
  scale_y_continuous( breaks = seq( from = 50, to = 300, by = 50 )) +
  labs( x = "Ano de filmagem"
      , y = "Total de apresentações"
      , fill = "Mês do ano"
      , title = "Publicações por mês em cada ano" ) +
  theme_bw()
```



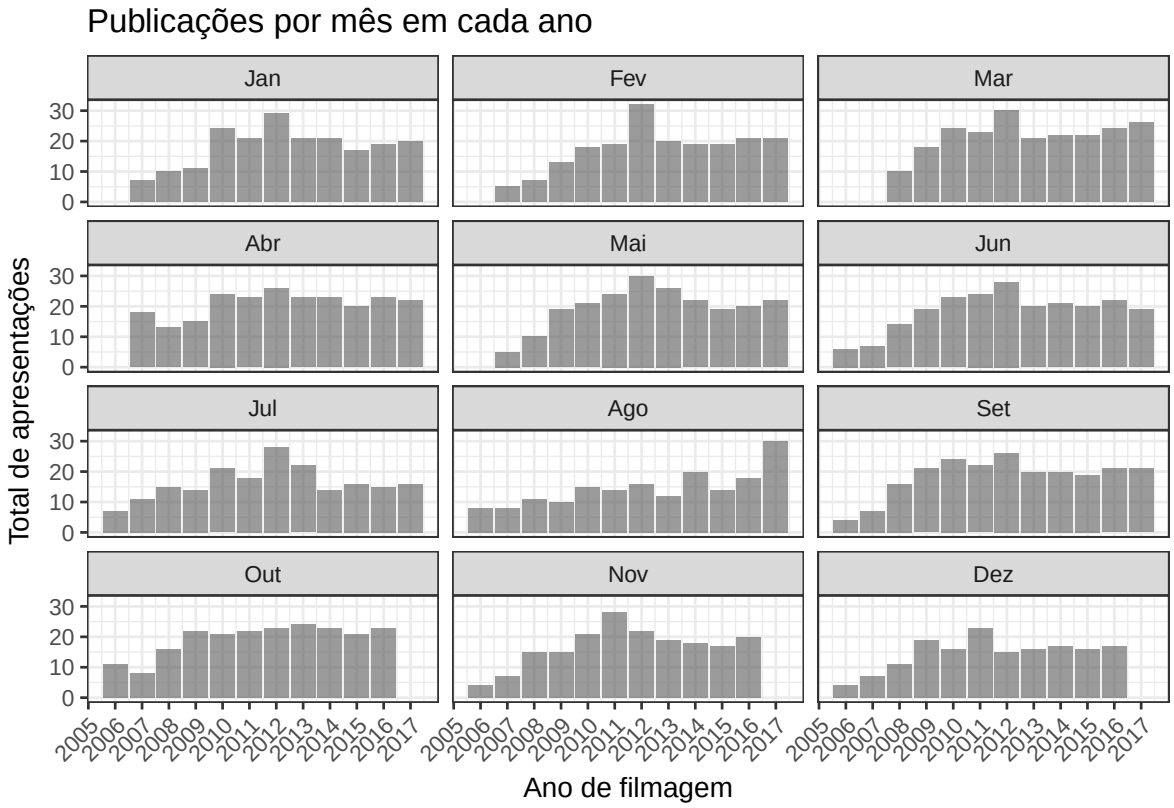
Facetas

Facetas são um recurso que possibilita a divisão de um canvas em vários gráficos homogêneos, separados por uma variável do data frame.

Neste exemplo repetimos o gráfico de barras original utilizando a função **facet_wrap**. Por meio desta função temos agora um gráfico por mês, organizados em um grid 3 x 4.

Ainda, o tema foi modificado para que o texto do eixo **x** seja exibido em um ângulo de 45 graus, viabilizando a exibição de todos os anos do intervalo.

```
ted_talks_recentes %>%
  mutate( ano = year( published_date ), mes = month( published_date, label = TRUE )) %>%
ggplot(aes( x = ano )) +
  geom_bar( alpha=0.6 ) +
  scale_x_continuous( breaks = 2005:2017 ) +
  facet_wrap (~ mes, ncol = 3 ) +
  labs( x = "Ano de filmagem"
      , y = "Total de apresentações"
      , fill = "Mês do ano"
      , title = "Publicações por mês em cada ano" ) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Boxplot

Boxplot é um tipo de gráfico que apresenta as relações de quartis de forma estruturada e contextualizada, além de indicar as faixas de valores.

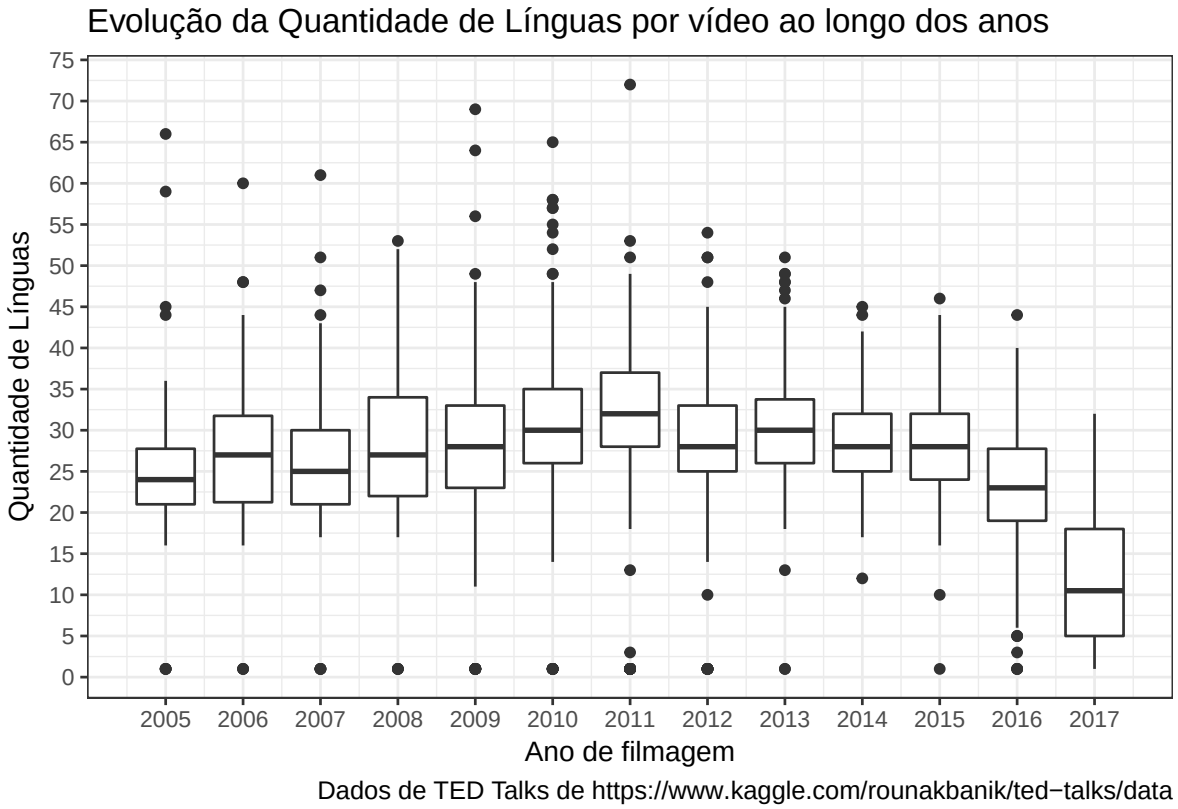
Como o ano é uma variável numérica, é necessário um mapeamento de estética para que a forma geométrica de boxplot compreenda como um grupo.

O boxplot apresenta as seguintes estatísticas:

- **Mediana**, a linha horizontal localizada dentro do retângulo.
- **Primeiro e terceiro quartis**, limites inferior e superior do retângulo.
- Relação Interquartil (**IQR**), onde:
  - a linha vertical superior alcança o maior valor menor ou igual a 1.5 * o IQR somado ao terceiro quartil
  - a linha vertical inferior alcança o menor valor maior ou igual a 1.5 * o IQR subtraído do primeiro quartil
- **Outliers**, que são quaisquer medidas que excedem as linhas de IQR

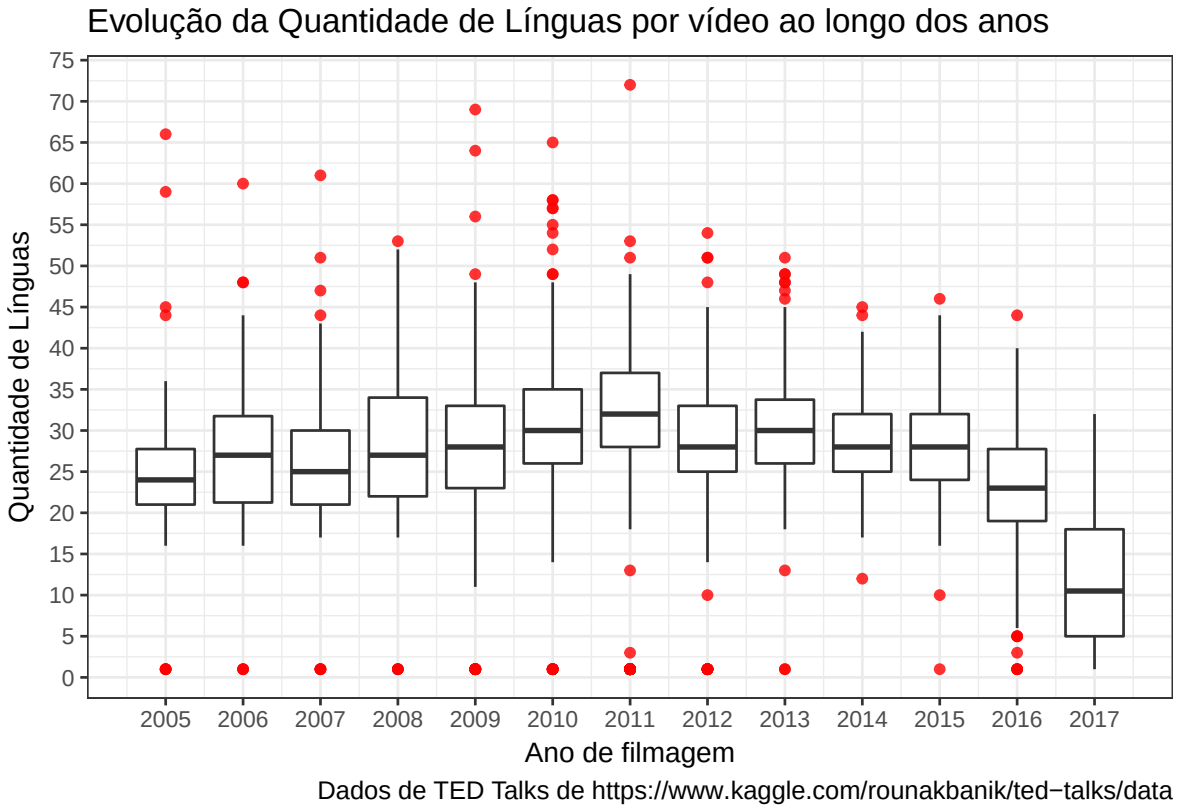
```
ted_talks_recentes %>%
  mutate( year = year( film_date )) %>%
ggplot( aes( x = year, y = languages, group = year )) +
  geom_boxplot() +
  scale_x_continuous( breaks = 2005:2017 ) +
  scale_y_continuous( breaks = seq(from = 0, to = 100, by = 5 )) +
  labs( x = "Ano de filmagem"
      , y = "Quantidade de Línguas"
      , title = "Evolução da Quantidade de Línguas por vídeo ao longo dos anos"
```

```
      , caption = "Dados de TED Talks de https://www.kaggle.com/rounakbanik/ted-talks/data") +
  theme_bw()
```



A forma geométrica de boxplot do ggplot2 possibilita customizar alguns dos componentes. No exemplo abaixo modifiquei a cor e a transparência dos pontos de outlier.

```
ted_talks_recentes %>%
  mutate( year = year( film_date )) %>%
ggplot( aes( x = year, y = languages, group = year )) +
  geom_boxplot(outlier.color = "red", outlier.alpha = 0.8) +
  scale_x_continuous( breaks = 2005:2017 ) +
  scale_y_continuous( breaks = seq(from = 0, to = 100, by = 5 )) +
  labs( x = "Ano de filmagem"
      , y = "Quantidade de Línguas"
      , title = "Evolução da Quantidade de Línguas por vídeo ao longo dos anos"
      , caption = "Dados de TED Talks de https://www.kaggle.com/rounakbanik/ted-talks/data") +
  theme_bw()
```

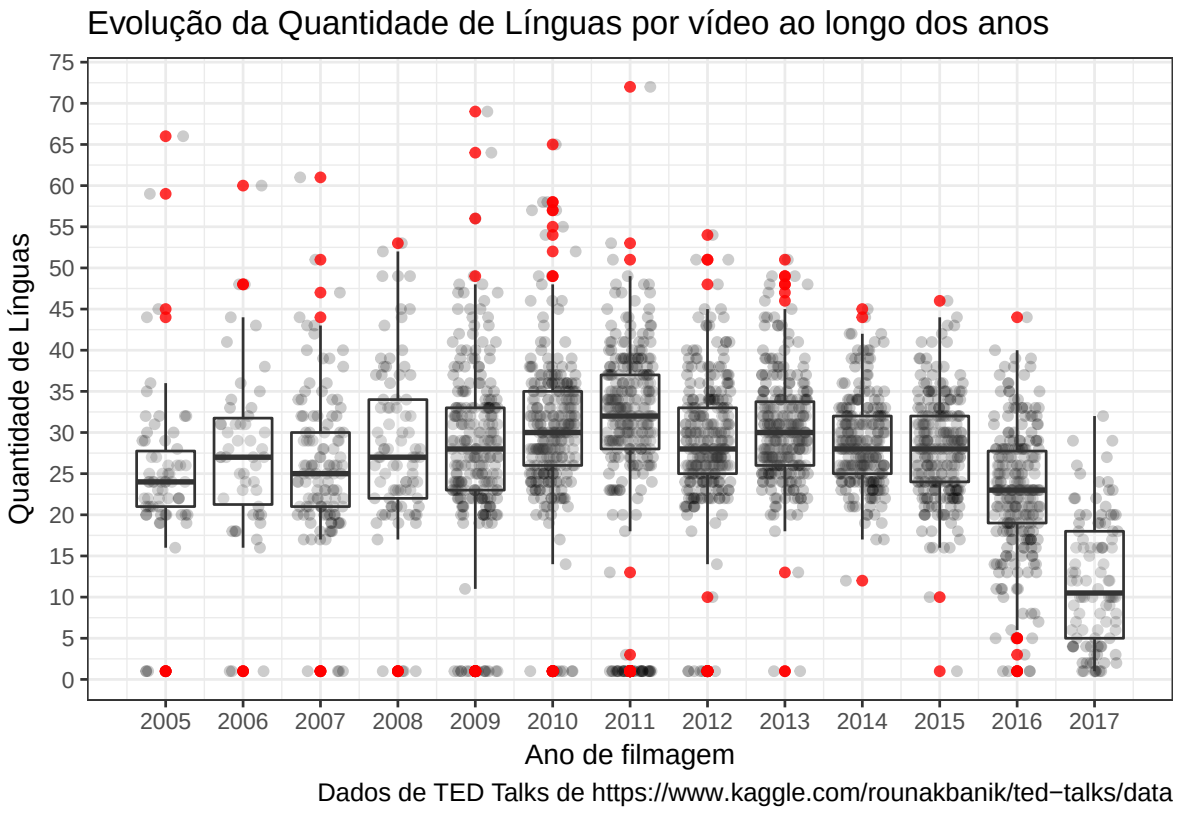


Composição com múltiplos gráficos

Assim como compomos gráficos combinando escalas, mapeamentos de estéticas e formas geométricas, podemos também combinar múltiplasa formas geométricas. O exemplo abaixo combina o boxplot com gráfico de pontos para ilustrar como as observações estão distribuídas.

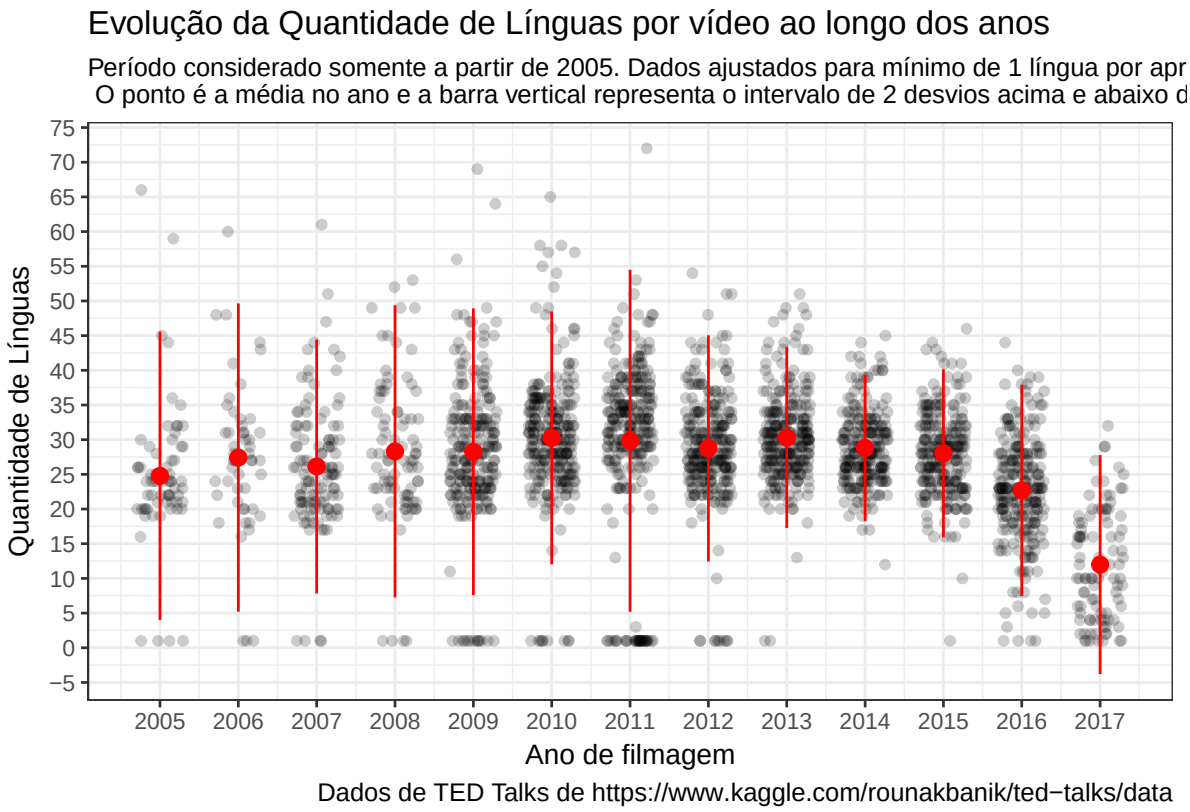
- **Jitter** (tremor) é uma variação de gráfico de pontos onde a posição é deslocada aleatoriamente em uma fração de altura e de largura. O exemplo abaixo aplica **jitter** para distribuir horizontalmente de forma que todas as observações estejam visíveis dentro de cada ano. Foi necessário aplicar uma transparência para melhor identificar a concentração
- As formas geométricas são sobrepostas na ordem em que são adicionadas ao canvas. No exemplo abaixo apliquei transparência no boxplot para que seja possível visualizar os pontos.

```
ted_talks_recentes %>%
  mutate( year = year( film_date )) %>%
ggplot( aes( x = year, y = languages, group = year )) +
  geom_jitter(alpha = .2, height = 0, width = 0.3) +
  geom_boxplot(outlier.color = "red", outlier.alpha = 0.8, alpha = 0.2) +
  scale_x_continuous( breaks = 2005:2017 ) +
  scale_y_continuous( breaks = seq(from = 0, to = 100, by = 5 )) +
  labs( x = "Ano de filmagem"
      , y = "Quantidade de Línguas"
      , title = "Evolução da Quantidade de Línguas por vídeo ao longo dos anos"
      , caption = "Dados de TED Talks de https://www.kaggle.com/rounakbanik/ted-talks/data") +
  theme_bw()
```



Combinando **jitter** com **stat_summary**

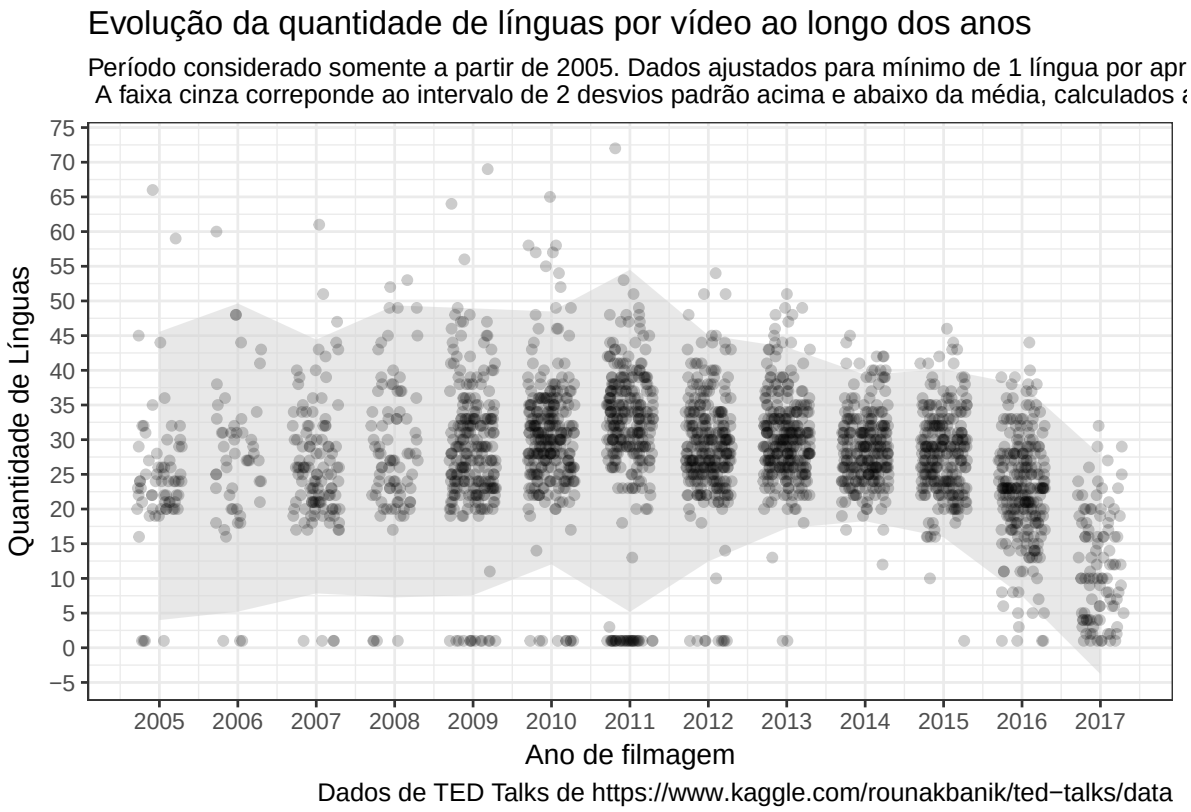
```
ted_talks_recentes %>%
  mutate( year = year( film_date )) %>%
ggplot( aes( x = year, y = languages )) +
  geom_jitter(alpha = .2, height = 0, width = 0.3) +
  stat_summary(fun.data = mean_sdl, color="red") +
  scale_x_continuous( breaks = 2005:2017 ) +
  scale_y_continuous( breaks = seq(from = -10, to = 80, by = 5 )) +
  labs( x = "Ano de filmagem"
      , y = "Quantidade de Línguas"
      , title = "Evolução da Quantidade de Línguas por vídeo ao longo dos anos"
      , subtitle = "Período considerado somente a partir de 2005. Dados ajustados para mínimo de 1 língu
      , caption = "Dados de TED Talks de https://www.kaggle.com/rounakbanik/ted-talks/data") +
  theme_bw()
```



Faixas de banda

A forma geométrica de banda é outra maneira de demarcar visualmente os limites superior e inferior através de estatísticas descritivas. Esta forma requer as estéticas **ymin** e **ymax**, que foram previamente calculadas no Data Frame.

```
ted_talks_recentes %>%
  mutate( year = year( film_date )) %>%
  group_by(year) %>%
  mutate(low = mean(languages) - 2 * sd(languages), hi = mean(languages) + 2 * sd(languages)) %>%
  ungroup() %>%
ggplot( aes( x = year, y = languages, ymin = low, ymax = hi )) +
  geom_ribbon(fill = "lightgray", alpha = 0.5) +
  geom_jitter(alpha = .2, height = 0, width = 0.3) +
  scale_x_continuous( breaks = 2005:2017 ) +
  scale_y_continuous( breaks = seq(from = -10, to = 80, by = 5 )) +
  labs( x = "Ano de filmagem"
      , y = "Quantidade de Línguas"
      , title = "Evolução da quantidade de línguas por vídeo ao longo dos anos"
      , subtitle = "Período considerado somente a partir de 2005. Dados ajustados para mínimo de 1 língu
      , caption = "Dados de TED Talks de https://www.kaggle.com/rounakbanik/ted-talks/data") +
  theme_bw()
```



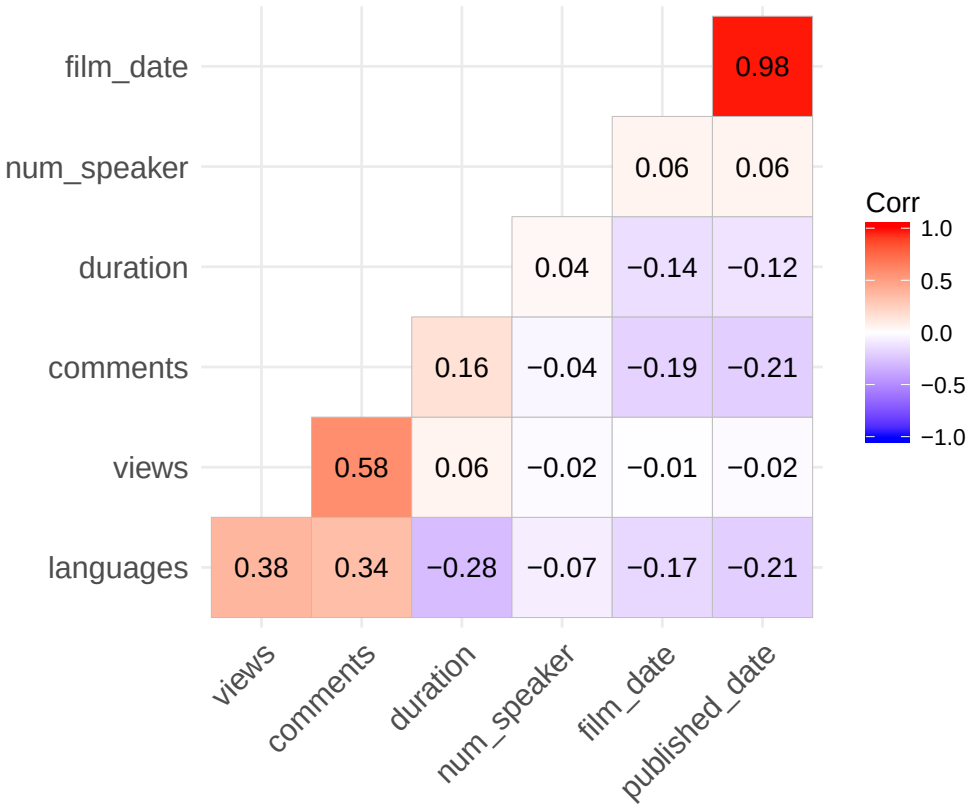
Correlograma

ggcorrplot possibilita visualizar as correlações entre variáveis numéricas. A matriz de correlações deve ser previamente calculada utilizando a função `cor`.

```
library(ggcorrplot)

corr <-
  ted_talks_recentes %>%
  select_if(is_numeric) %>%
  mutate( duration = as.numeric(duration)
        , published_date = as.numeric(published_date)
        , film_date = as.numeric(film_date)) %>%
  select(-event, -speaker_occupation) %>%
  cor() %>% round(2)

ggcorrplot(corr, hc.order = TRUE, type = "lower", lab = TRUE)
```



Histograma

ATIVIDADE

1. Estude o material abaixo que explica a construção de histogramas

- http://flowingdata.com/2017/06/07/how-histograms-work/
- http://tinlizzie.org/histograms/

2. Estude o help da função `geom_histogram`
3. Crie um histograma da quantidade de visualizações multifacetado por ano de publicação, restrito aos anos entre 2012 e 2017.

O resultado desta atividade deve ser um arquivo chamado "03-atividade-extra.R" dentro do diretório aula-05. O script em R deve carregar em um Data Frame o conteúdo do arquivo de dados das TED Talks e criar os histogramas de forma multifacetada, conforme apresentado neste material de aula.

Você deve publicar o arquivo .R no Github para avaliação.

FIM ATIVIDADE